Elliptic Curve Cryptography and Its Future

Joseph Patrick C. Galvez

American Military University

ISSC266: Cryptography Concepts

Mr. Joel Stewart

Due Date

## I. Introduction

Ever since computers were connected to a network and information needed to be transmitted between them there existed a need to keep those communications confidential; in fact, the first computer network to use packet switching was ARPANET, which was created by ARPA, a DoD subsidiary. However, even before the advent of the modern internet algorithms and cryptosystems existed to ensure the secure exchange of information. RSA and the Diffie-Hellman key exchange were developed and adopted in the 1970s as the first asymmetric key exchange algorithms, in the following decade ECC or Elliptic Curve Cryptography was developed which provided better security than RSA whilst using a smaller key size. The security of ECC is a result of the Elliptic Curve Discrete Logarithm Problem which utilizes predefined geometries provided by the NIST as a scaffold for its key generation and validation operations. ECC is used in the Elliptic-Curve Diffie-Hellman and the Elliptic-Curve Digital Signature Algorithm

## II. Mathematical Foundations of ECC

Elliptic Curve Cryptography depends on the Elliptic Curve as the scaffold of its mathematical operations. The two types of curves can be described as binary curves[[1]](#footnote-1) represented by the eqution and non-binary curves, aka prime curves, which can be expressed by the equation (Chen et al 2023, ch.2). The curves recommended for use in NIST SP 800-186 have certain criteria the first being that the underlying finite field must either be a prime field or be defined by where m is a prime number(Chen et al 2023, appendix c.1). The second being that the curve group is defiend by |E| = h⋅n which determines every possible point on the curve. The third is that the base point of G on the curve will be the fixed and publicly agreed-upon starting point of all point generation. The fourth is that the size of the finite curve cannot be the size of the entire curve, this is done to prevent the creation of an anomalous curve which compromises the security of ECC as this makes it vulnerable to a smart attack[[2]](#footnote-2). The fifth criteria for there to be a large embedding degree where k is at least or 1024 in order to prevent the EDCLP being reduced to a simpler problem. Lastly, the discriminant is used to describe the relationship between the curve and its endomorphism. Regardless of the equation used elliptic curves must be capable of group law visualization. NIST Special Publication 800-186 cites the group laws for different types of curves, those being the Short-Weierstrass curve and the Twisted Edwards Curve; among them the Weierstrass curve remains the most commonly used.

The foundation of ECC is the Elliptic Curve Discrete Logarithm Problem. In brief one of the goals of ECDLP is to seemingly create random numbers for use in secret key generation by using scalar multiplications on the curve. The geometry of the curve is determined by the Generator point and subsequent points were placed on the curve by means on point addition and scalar multiplication. Among the predetermined geometry of the curve a point is chosen by the secret integer to act as the private key. This process is described in NIST SP 800-56A in section 5.6.1.2,” each valid public key Q is related to the corresponding (valid) private key d by the following formula: Q = () = dG. “; here the relationship between the Generator point and the secretkey are shown to corelate with the public key. Centicom Research (2000) describes the use of scalar multiplication in point generation in section 1-2,” Given an integer k and a point P E ( ), scalar multiplication is the process of adding P to it”. A more holistic explanation of ECDLP is stated plainly in a commentary on section 2,” The ECDLP is, given E, G, and a scalar multiple Q of G, to determine an integer l such that Q = lG.”, This lays bare the importance of scalar multiplication in the difficulty of ECDLP in the creation of the unknown integer. ECDLP creates the foundation of ECC by creating a problem that is computationally efficient, thanks to the geometry of ECC and difficult to solve without the secret integer.

## III. Practical Applications of ECC

The Elliptic Curve Logarithm Problem is the foundation of ECC, as such it defines the security of ECC protocols like Elliptic Curve Diffie-Hellman which is responsible for secure public key exchange. ECDH requires three things to securely exchange a key: domain parameters, the private key, and the public key(NIST, 2018, Section 5.7.1.2). The first step of ECDH is to compute where P is the shared point on the curve, is the public key, is the private key, and h is the cofactor; in other words, the shared point is the product of the cofactor and the private and public keys resulting from scalar multiplication. Next, if then all values used to arrive to P are deleted because in the case P is equal to the identity of the curve, which is a clear sign of an error or an invalid key. Next, if P is valid and does not equal the indentity of the curve the raw x value of P is extracted and is converted into Z which is the actual shared secret used in ECDH. This is done through field-element-to-byte string conversion. Next the intermediate results used to arrive to Z are destroyed, finally Z is outputted. In this way efficient and secure key exchanges are made possible through the use of the eiiliptic curve geometry.

Another Cryptographic Protocol that uses ECDLP as its foundation is the Elliptic Curve Digital Signature Algorithm. ECDSA needs four things to work: M – the bit string to be signed, the private key -d, the domain parameters – D, and an approved hash function(NIST, 2023, Section 6.4). First a hash (H) is computed where the hashing algorithm creates a hash of the bit string. Next, an integer (e) is derived from the hash. Next, a secret number is generated for each message (data to be signed), the

value of k will be within 0 < k < n. Next is computed to find the modular inverse of k. Point R is then derived as the scalar multiplication product of [k]G. The following steps simply ensure that R is in an affinine format meaning that the format of the coordinates on the curve look like (). Next the x coordiante is converted into an integer . Next, r is computed from through the equation mod n. Next s is computed through the equation , this operation is not scalar. K and are then destroyed. Finally (r,s) is outputted and if either r or s are 0 then it is a clear sign that computation was a failure, if so a new secret number is used and new computations are attempted. Although r is easy to understand as the integer version of the x coordinate of the curve s is a computation made from the private key, the hash message and r. In this way the two integers act as a signature that both work together to verify the authenticity of a message and the integrity of the data.

Both Elliptic-Curve Diffie-Hellman and Elliptic-Curve DSA are in common uses throughout various applications that utilize them for the purposes of key exchange and for digital signatures. An example of a practical application of ECDH and EDCSA is in the transmission of credit card cvc information according to NIST SP 800-78-4,” The secure messaging CVC shall be signed using ECDSA (Curve P-256) with SHA-256 if it contains an ECDH (Curve P-256)”.Furthermore, ECDH was outlined as an approved algorithm for use in TLS 1.2 specifically in the cipher suite TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA in NIST SP 800-52(NIST, 2018, 3.3.1). ECDH is also used as a protocol in secure emails, specifically it is used in CMS, the algorithm identifier ecdsa-with-SHA256 is specifically mentioned in RFC 5753 (Turner & Brown, 2010). Another application that benefits from ECC is secure Bluetooth device connection,” The Secure Connections protocol itself is a protocol family, all members sharing an Elliptic-Curve Diffie–Hellman (ECDH) key exchange with key confirmation”( Fischlin & Sanina, 2021, p.3). The wide range of applications that use ECDSA and ACDH demonstrate how ECC is desirable due to both its security and computational efficiency, as such is has cemented its current position in securing the information exchange present in our daily lives.

## IV. Future trends in the area

With ECC being as prominent as it is many companies have adopted it in their operations, some are even actively involved in its development, among them are some of the most valuable companies on the planet. Although the development of ECC was largely sponsored by government organizations such as the NSA and the NIST, the private sector has also made its contributions. Centricom is one such company, its contribution was the first publicly available ECC toolkit which helped simplify and improve ECC (Certicom, n.d). In addition to improving ECC other companies can also provide the support necessary to keep it working such as Entrust which provides digital certificates and PKI services. The development and support provided by the aforementioned companies allow larger companies to provide secure services en masse. Apple, for example, uses ICIES as its encryption scheme for the ubiquitous message service, imessage(Apple Inc., 2022). Google and other browsers use TLS which depends on ECC for secure browsing protocols such as HTTPS. ECC’s widespread use is observed in its use by some of the world’s most prominent IT companies as well as the continued investment by others.

Given its widespread use it makes sense that ECC has established regulations, the main regulatory body governing the use of ECC is the NIST which has published the FIPS series to provide specifications on its application and standards for use on federal systems. One such publication is FIPS 186-5 which determines the accepted geometries of the Elliptic Curves meant to be used for ECC (NIST, 2023). NIST PS 800-56A gives outlines of schemes for use in ECC and gives the rationale on which schemes to select which gives entities a useful reference in implementing ECC. As such, the NIST acts as the main regulatory body in regards to ECC and provides regulatory guidance in regards to implementation and operation.

Considering the reach of ECC and the reliance it has incurred it is important to note how ECC might be affected by quantum computing as well as how it may evolve. Moody(2019) claims that both ECDSA and ECDH are vulnerable to quantum attacks specifically they are vulnerable to Shor’s algorithm. The recommended improvement to guard against quantum threats is to add new curves which are named Ed25519 and Ed448. The final slide of the presentation outlines that a transition to a new algorithm can happen in 10 years from publication.

## V. Conclusion

In conclusion, ECC is a subset of digital cryptography which utilizes predefined geometries known as elliptic curves, with some conditions defined in NIST SP 800-186. The geometry is predefined and makes it computationally efficient to define all available points for key generation. Although the process to create a secret key is streamlined it is very difficult to find the secret key with just the public key and the curve geometry, this is the problem that lies in the core of ECC and is known as the Elliptic Curve Discrete Logarithm Problem. The difficulty of ECDLP is the underlying security assumption in ECC protocols such as Elliptic Curve Diffie-Hellman and the Elliptic Curve Digital Signature Algorithm which are then utilized by the world’s largest companies to secure everything from secure messaging to e-commerce. Lastly, ECC is regulated by the NIST, which has also outlined solutions for guarding ECC against vulnerabilities that arise from quantum computing, especially those that utilize Shor’s Algorithm.

**References**

Chen, L., Moody, D., Randall, K., Regenscheid, A., & Robinson, A. (2023). *Recommendations for Discrete Logarithm‑based Cryptography: Elliptic Curve Domain Parameters* (NIST Special Publication 800‑186). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-186>

Barker, E., Chen, L., Roginsky, A., & Smid, M. (2013). Recommendation for Pair‑Wise Key‑Establishment Schemes Using Discrete Logarithm Cryptography (NIST Special Publication 800‑56A Rev. 2). National Institute of Standards and Technology. Retrieved June 25, 2025, from <https://csrc.nist.gov/csrc/media/publications/sp/800-56a/rev-2/final/documents/draft-sp-800-56a.pdf>

**National Institute of Standards and Technology. (2018). *Recommendation for Pair‑Wise Key‑Establishment Schemes Using Discrete Logarithm Cryptography (SP 800‑56A Rev. 3)***. NIST. <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/nist.sp.800-56Ar3.pdf>

Certicom Research. (2000, September 20). *SEC 1: Elliptic Curve Cryptography (Version 1.0)*. Standards for Efficient Cryptography Group. <https://www.secg.org/SEC1-Ver-1.0.pdf>

**National Institute of Standards and Technology. (2023). *Digital Signature Standard (DSS) (FIPS 186‑5)*.** NIST. <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5.pdf>

Polk, T., Dodson, D., Burr, W. E., Ferraiolo, H., & Cooper, D. (2015).  
*Cryptographic algorithms and key sizes for personal identity verification* (NIST Special Publication 800‑78‑4). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-78-4>

**National Institute of Standards and Technology.** (2018, October 15). *Guidelines for the selection, configuration, and use of Transport Layer Security (TLS) implementations (NIST Special Publication 800‑52 Revision 2: Draft).* <https://csrc.nist.gov/CSRC/media/Publications/sp/800-52/rev-2/draft/documents/sp800-52r2-draft.pdf>

**Kampanakis, P., & Childs‑Klein, W.** (2024, March). *The impact of data‑heavy, post‑quantum TLS 1.3 on the time‑to‑last‑byte of real‑world connections* [Paper presentation]. Fifth PQC Standardization Conference, National Institute of Standards and Technology.

Turner, S., & Brown, D. R. L. (2010, January). *Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)* (RFC 5753). RFC Editor. <https://doi.org/10.17487/RFC5753>

Fischlin, M., & Sanina, O. (2021). *Cryptographic analysis of the Bluetooth Secure Connection Protocol Suite* (Cryptology ePrint Archive, Paper 2021/1597). <https://doi.org/10.1007/978-3-030-92075-3_24>

**Certicom (BlackBerry)**. (n.d.). *Elliptic Curve Cryptography (ECC)*. Retrieved June 26, 2025, from <https://www.certicom.com/content/certicom/en/ecc.html>

Apple Inc. (2022, May 13). *How iMessage sends and receives messages securely*. In *Apple Platform Security*. Retrieved June 26, 2025, from <https://support.apple.com/guide/security/how-imessage-sends-and-receives-messages-sec70e68c949/web>

National Institute of Standards and Technology. (2023, February 3). *Digital Signature Standard (DSS) (FIPS Publication 186‑5)*. U.S. Department of Commerce. <https://doi.org/10.6028/NIST.FIPS.186-5>

Moody, D. (2019, March). *NIST status update on elliptic curves and post‑quantum crypto* [Presentation]. NIST Threshold Cryptography Workshop. National Institute of Standards and Technology. <https://csrc.nist.gov/CSRC/media/Presentations/NIST-Status-Update-on-Elliptic-Curves-and-Post-Qua/images-media/moody-dustin-threshold-crypto-workshop-March-2019.pdf>

1. Binary Curves are primarily used in smart cards if at all today. [↑](#footnote-ref-1)
2. Also known as an anomalous curve attack. [↑](#footnote-ref-2)